Основные нововведения Java 8 это лямбда-выражения, stream и изменения в интерфейсе программирования (API). Рассмотрим некоторые из них:

1. Более короткий синтаксис за счёт применения лямбда-выражений.

Лямбда выражение имеет структуру (параметры) **->** {тело метода}, где

*параметры* – список входящих параметров разделённых через запятую

**->** - оператор, который разделяет параметры от тела метода

*тело метода* – одно или несколько выражений;

1. Пакеты *java.util.stream* , который применяется для работы с потоками используются пакет *java.util.stream*. Stream — это объект для универсальной работы с данными. Требуется указать только операции, которые хотим произвести без вникания в датели, например

*public static void main(String[] args) {*

*List<String> list = Arrays1.stream(args)*

*.filter(s -> s.length() <= 10)*

*.collect(Collectors.toList());*

*}*

В данном примере источником исходных данных является метод  Arrays1.stream, который из массива args делает stream. Промежуточный оператор filter отбирает только те строки, длина которых не превышает 10-и. Терминальный оператор collect собирает полученные элементы в новый список.

1. В *java.util.function* содержаться ряд функциональных интерфейсов для поддержки лямбда-выражений.
2. Возможность неабстрактной реализации по умолчанию метода, объявленного в интерфейсе. То есть если конкретная реализация метода отсутствует, то используется его реализация по умолчанию, например

*interface Equal {*

*double calculate(int a);*

*default double sqrt(int a) {*

*return Math.sqrt(a); }*

*}*

В данном примере интерфейс Equal определяет метод по умолчанию sqrt .

1. Новый прикладной программный интерфейс для обработки даты и времени, типовые аннотации и возможность параллельно обрабатывать массив.

Основные отличия Java 9:

1. Полная поддержка клиента HTTP 2.0. HTTP 2.0 предоставляет более высокие результаты, скорости передачи данных по сравнении с клиентом HTTP 1.1.
2. Jshell: Новый инструмент командной строки. Если разработчик хочет автономно запустить несколько строк Java, то это можно выполнить без необходимости заворачивать все в отдельный метод или проект.
3. Microbenchmark: Теперь производительность отдельных небольших частей кода можно измерить стандартизированным методом.

Основные отличия Java 10:

1. Вывод типа локальной переменной: улучшает язык Java, позволяя сократить объявление переменной до ключевого слова var.
2. Parallel Full GC для G1: усовершенствованный сборщик мусора.
3. Application Class-Data Sharing: ряд наработок, позволяющих классам приложений размещаться в одном архиве.
4. Экспериментальный JIT-компилятор на базе Java: новый JIT-компилятор Java VM.

Основные отличия Java 11:

1. Новые API;
2. Удалены модули Java EE и COBRA;
3. var-синтаксис для лямбда-параметров;

Основные отличия Java 12:

1. Новая форма написания выражения с оператором switch. Например

*switch (day) { case MONDAY, FRIDAY, SUNDAY -> System.out.println(6)* (без break);

1. Обновления сборщика мусора G1;
2. Ускорение загрузки за счёт CDS (Class Data Sharing). Данная функция упаковывает все запускаемые при старте классы в специальный class data sharing архив, используя по список классов по умолчанию.

Основные отличия Java 13:

1. Изменения синтаксиса языка за счёт текстовых блоков;
2. Использование switch как выражения, возвращающего результат, который можно присвоить переменной;
3. Новые API.

Список используемых материалов:

<https://www.oracle.com/>

<https://javarush.ru/>

<https://www.ibm.com/>

<https://habr.com/>